This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) An apparatus for providing a user interface to a designer of documents, said apparatus comprising:

an input for receiving input from said designer, said input comprising a first event-driven ~~programs~~ program, said first event-driven program coding a method or property of an object;

a converter converting said first event-driven program ~~programs~~ into serial execution code including a markup language, wherein said first event-driven program presents a visual representation of said object based on said serial execution code to said designer;

an output for outputting said serial execution code to a server that serially executes said serial execution code,

upon command by said designer, said converter reconverting said serial execution code into a second event-driven ~~programs~~ program, wherein said second event-driven program presents a visual representation of said object based on the executed serial execution code to said designer.

2. (Original) The apparatus of claim 1, further comprising:

a client connected to said server, said client receiving the output of said serial execution code;

wherein the user interface provided to said designer displays the programs that operate between said client and server as programs that operate as a single machine.

3. (Original) The apparatus of claim 1, wherein said event-driven programs include objects.

4. (Original) The apparatus of claim 3, further comprising: a script library for storing a script relating to objects for later placement in said first event-driven programs.

5. (Previously presented) The apparatus of claim 3, said apparatus further comprising:

design-time controls for controlling the generation of said objects when said design-time controls are placed within said first event-driven programs.

6. (Original) The apparatus of claim 1, wherein said first and said second event driven programs are the same event-driven programs.

7. (Original) The apparatus of claim 1, wherein said first and second event driven programs are different event-driven programs.

8. (Currently amended) A method for operating with a user interface provided to a designer of documents, said user interface representing documents as event-driven, said method comprising the steps of:

receiving an input from said designer, said input comprising a first event-driven ~~programs~~ program, said first event-driven program coding a method or property of an object;

converting said first event-driven ~~programs~~ program into serial execution code including a markup language, wherein said first event-driven program presents a visual representation of said object based on said serial execution code to said designer;

outputting said serial execution code to a server that serially executes said serial execution code,

upon command by said designer, reconverting said serial execution code into a second event-driven ~~programs~~ program wherein said second event-driven program presents a visual representation of said object based on the executed serial execution code to said designer.

9. (Original) The method of claim 8, further comprising the step of:

receiving the output of said serial execution code at a client connected,

wherein the user interface provided to said designer displays the programs that operate between said client and server as programs that operate as a single machine.

10. (Original) The method of claim 8, wherein said event-driven programs include objects.

11. (Original) The method of claim 10, further comprising the step of:

storing in a script library a script relating to objects for later placement in said first event-driven programs.

12. (Previously presented) The method of claim 10, further comprising the steps of:

controlling the generation of said objects with controls that operate during a design time when said controls are placed within said first event-driven programs.

13. (Original) The method of claim 8, wherein said first and said second event driven programs are the same event-driven programs.

14. (Original) The method of claim 8, wherein said first and second event driven programs are different event-driven programs.

15. (Previously presented) The apparatus according to claim 1, wherein said markup language includes hypertext markup language.

16. (Previously presented) The method according to claim 8, wherein said markup language includes hypertext markup language.

17. (Withdrawn) A computer system for designing internet-accessible datasets comprising:

a processor;

a first storage that, in combination with said processor, provides a design space to a developer where the developer develops programs that call objects with methods and properties,

wherein said processor converts said programs from said design space into a runtime space in which said objects with methods and properties are represented as server-executable web pages and where a first page of said pages invokes a method or property from a second page of said pages.

18. (Withdrawn) The computer system according to claim 17, wherein said design space is an event-driven space.

19. (Withdrawn) The computer system according to claim 18, wherein said runtime space exists on a server that processes said web pages.

20. (Withdrawn) The computer system according to claim 18, wherein said runtime space becomes an interaction space when a remote client interacts with said server-executable web pages served by a server.